

[illegible]

HH	HH	EEEEEEEEEE	AAAAAA	DDDDDDDD	EEEEEEEEEE	RRRRRRRR	
HH	HH	EEEEEEEEEE	AAAAAA	DDDDDDDD	EEEEEEEEEE	RRRRRRRR	
HH	HH	EE	AA	DD	EE	RR	RR
HH	HH	EE	AA	DD	EE	RR	RR
HH	HH	EE	AA	DD	EE	RR	RR
HH	HH	EE	AA	DD	EE	RR	RR
HH	HH	EEEEEEEEEE	AA	DD	EEEEEEEEEE	RRRRRRRR	
HH	HH	EEEEEEEEEE	AA	DD	EEEEEEEEEE	RRRRRRRR	
HH	HH	EE	AAAAAAAAAA	DD	EE	RR	RR
HH	HH	EE	AAAAAAAAAA	DD	EE	RR	RR
HH	HH	EE	AA	DD	EE	RR	RR
HH	HH	EE	AA	DD	EE	RR	RR
HH	HH	EEEEEEEEEE	AA	DDDDDDDD	EEEEEEEEEE	RR	RR
HH	HH	EEEEEEEEEE	AA	DDDDDDDD	EEEEEEEEEE	RR	RR

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLLL	IIIIII	SSSSSSSS

```
0001 0
0002 0 MODULE HEADER (LANGUAGE (BLISS32) ,
0003 0 IDENT = 'V04-000' ,
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: MTAACP
0033 1
0034 1 ABSTRACT:
0035 1 This module contains routines which position to headers or trailers
0036 1 and read them.
0037 1
0038 1 ENVIRONMENT:
0039 1
0040 1 Starlet operating system, including privileged system services
0041 1 and internal exec routines.
0042 1
0043 1 --
0044 1
0045 1
0046 1
0047 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 25-MAY-77 15:00
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1 V03-006 MMD0323 Meg Dumont, 13-Aug-1984 15:17
0052 1 Fix to fix MMD0285, the way it was implemented the call
0053 1 wasn't getting made.
0054 1
0055 1 V03-005 MMD0300 Meg Dumont, 20-Jun-1984 11:23
0056 1 Fix to default Buffer offset length to zeros, when no HDR2
0057 1 is present for the file.
```



```

58 0058 1
59 0059 1
60 0060 1
61 0061 1
62 0062 1
63 0063 1
64 0064 1
65 0065 1
66 0066 1
67 0067 1
68 0068 1
69 0069 1
70 0070 1
71 0071 1
72 0072 1
73 0073 1
74 0074 1
75 0075 1
76 0076 1
77 0077 1
78 0078 1
79 0079 1
80 0080 1
81 0081 1
82 0082 1
83 0083 1
84 0084 1
85 0085 1
86 0086 1
87 0087 1
88 0088 1
89 0089 1
90 0090 1
91 0091 1
92 0092 1
93 0093 1
94 0094 1
95 0095 1
96 0096 1
97 0097 1
98 0098 1
99 0099 1
100 0100 1
101 0101 1
102 0102 1
103 0103 1
104 0104 1
105 0105 1
106 0106 1
107 0107 1
108 0108 1
109 0109 1
110 0493 1
111 0494 1
112 0495 1
113 0496 1
114 0497 1

V03-004 MMD0285 Meg Dumont, 6-Apr-1984 17:18
Fix to READ_HDR to include calling the clear
serious exception routine after the headers are
read. This is so that we do not leave the
TMSCP drives left in serious exception state
if we read into the TM while reading the headers.

V03-003 MMD0280 Meg Dumont, 23-Mar-1984 10:27
Fix long file name support such that for ANSI version
3 volumes it converts the exentsion length to
ASCII characters before writing it to the label.

V03-002 ROW0258 Ralph O. Weber 21-NOV-1983
The Paul Painter Memorial Enhancement
Named for one of the unfortunate customers who suffered much
to determine the great UCBSL_MT_RECORD secret while trying to
create a user-written magtape driver, this change eliminates
use of the device dependent field, UCBSL_MT_RECORD in favor of
the device independent field, UCBSL_RECORD.

V03-001 MMD0162 Meg Dumont, 26-Apr-1983 9:36
Change reference to 80 to the symbol ANSI_LBLSZ. Change READ_HDR
to read in the HDR4 label or if not found to default the values.

V02-010 REFORMAT Maria del C. Nasr 30-Jun-1980

V02-009 MCN0016 Maria del C. Nasr, 18-Jun-1980 11:55
Initialize default HDR2 with blanks, instead of zeroes, to
avoid setting the old RMS attributes field.

A0008 MCN0013 Maria del C. Nasr 11-Mar-1980 11:25
Check for HDR3 instead of HDR2 to determine if current file
should be included in search or not.

A0007 MCN0011 Maria del C. Nasr 04-Feb-1980 9:05
Add input parameter to UPDVCB_LEOV routine to either clear
or set flag, and make routine global.

A0006 MCN0003 Maria del C. Nasr 28-Sep-79 10:39
Add HDR3 processing

A0005 SPR24948 Maria del C. Nasr 11-Sep-79 17:30
Forced spacing to eof when current position bit set to
fix bug.

**
LIBRARY 'SYSS$LIBRARY:LIB.L32';
REQUIRE 'SRC$;MTADEF.B32';

FORWARD ROUTINE
READ_HDR : COMMON_CALL, ! read HDR1, HDR2, and HDR3 and HDR4 if exist
SPACE_EOF : COMMON_CALL NOVALUE, ! space to end of file
SET_NUMBER_OF_LABELS : COMMON_CALL NOVALUE, ! set the number of labels read

```

HEADER
V04-000

D 1
16-Sep-1984 02:22:07
14-Sep-1984 12:46:41

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]HEADER.B32;1

Page 3
(1)

: 115	0498	1	UPDVCB LEQV	: COMMON_CALL NOVALUE,	! update VCB logical end of file
: 116	0499	1	MAKE_COR_FILE	: COMMON_CALL NOVALUE,	! update VCB
: 117	0500	1	WRAP_AROUND	: L\$WRAP_AROUND;	! continue search at beginning of volume set !
: 118	0501	1			
: 119	0502	1	EXTERNAL		
: 120	0503	1	CURRENT_UCB	: REF BBLOCK,	
: 121	0504	1	IO_PACKET	: REF BBLOCK,	! address of IO request packet
: 122	0505	1	HDR1	: REF BBLOCK,	! address HDR1 label
: 123	0506	1	HDR2	: REF BBLOCK,	! address of HDR2 label
: 124	0507	1	HDR3	: REF BBLOCK,	! address of HDR3 label
: 125	0508	1	HDR4	: REF BBLOCK,	! address of HDR4 label
: 126	0509	1			

```

128 0510 1 GLOBAL ROUTINE GET_START_HDR : L$GET_START_HDR =
129 0511 1
130 0512 1 !++
131 0513 1
132 0514 1 FUNCTIONAL DESCRIPTION:
133 0515 1 This routine positions to the header label set of the start file
134 0516 1 in current search and reads HDR1, HDR2, HDR3 and HDR4 labels unless
135 0517 1 they have already been read.
136 0518 1
137 0519 1 CALLING SEQUENCE:
138 0520 1 GET_START_HDR()
139 0521 1
140 0522 1 INPUT PARAMETERS:
141 0523 1 none
142 0524 1
143 0525 1 IMPLICIT INPUTS:
144 0526 1 CURRENT_VCB, CURRENT_UCB
145 0527 1
146 0528 1 OUTPUT PARAMETERS:
147 0529 1 none
148 0530 1
149 0531 1 IMPLICIT OUTPUTS:
150 0532 1 HDR1 read in, HDR2 read in or defaulted, HDR3 read in or defaulted
151 0533 1 HDR4 read in or defaulted
152 0534 1
153 0535 1 ROUTINE VALUE:
154 0536 1 0 unsuccessful, logical end of volume set
155 0537 1 1 successful
156 0538 1
157 0539 1 SIDE EFFECTS:
158 0540 1 none
159 0541 1
160 0542 1 --
161 0543 1
162 0544 2 BEGIN
163 0545 2
164 0546 2 EXTERNAL REGISTER
165 0547 2 COMMON_REG;
166 0548 2
167 0549 2 EXTERNAL ROUTINE
168 0550 2 MOUNT_VOL : COMMON_CALL; ! mount volume
169 0551 2
170 0552 2 EXTERNAL
171 0553 2 CURRENT_UCB : REF BBLOCK, ! address of current ucb
172 0554 2 LOCAL_FIB : BBLOCK; ! copy of user's fib
173 0555 2
174 0556 2 LOCAL
175 0557 2 RELATIVE_BLOCK, ! relative block number to last tm
176 0558 2 TM; ! number of tm's
177 0559 2
178 0560 2 ! mount volume if the current relative volume number is zero
179 0561 2 !
180 0562 2
181 0563 2 IF .CURRENT_VCB[VCBSB_CUR_RVN] EQL 0
182 0564 2 THEN
183 0565 2 MOUNT_VOL(1, $FIELDMASK(MOUSV_REWIND) + $FIELDMASK(MOUSV_LBLCHECK));
184 0566 2

```



```

185 0567 2 ! if at logical end of volume set, return immediately
186 0568
187 0569
188 0570 IF .CURRENT_VCB[VCBSV_LOGICEOVS]
189 0571 THEN
190 0572     RETURN 0;
191 0573
192 0574 ! If the number of tape marks into the file is not 0, then the previous file
193 0575 ! was closed prematurely and should not be included in search except in the
194 0576 ! case where there is no HDR3 and the tape is left positioned beyond the
195 0577 ! tm. If the section is not the first, then space to next file
196 0578
197 0579
198 0580 IF (.CURRENT_VCB[VCBSB_TM] NEQU 0
199 0581     AND
200 0582     NOT (.CURRENT_VCB[VCBSB_TM] EQLU 1 AND .HDR3[HD3$SL_HD3LID] NEQU 'HDR3'
201 0583     AND
202 0584     (.CURRENT_UCB[UCBSL_RECORD] - .CURRENT_VCB[VCBSL_ST_RECORD]) EQLU 0))
203 0585     OR
204 0586     .CURRENT_VCB[VCBSW_CUR_SEQ] GTR 1
205 0587 THEN
206 0588     SPACE_EOF() ! position to beginning of next file
207 0589 ELSE
208 0590
209 0591 ! If function is create, and current position bit is set, then force
210 0592 ! spacing to end of file, unless positioned in dummy file header set...
211 0593
212 0594
213 0595 IF ((.IO_PACKET[IRPSV_FCODE] EQL IO$_CREATE) AND .LOCAL_FIB[FIBSV_CURPOS])
214 0596     AND
215 0597     (.CURRENT_VCB[VCBSB_TM] NEQU 0) AND (.CURRENT_VCB[VCBSW_CUR_NUM] NEQU 0)
216 0598 THEN
217 0599     SPACE_EOF();
218 0600
219 0601 ! When new volume is mounted, VOL1 has been read but not the header labels.
220 0602 ! Therefore, the actual block count equals 1. If relative block count = 0,
221 0603 ! then the headers have not been read for this file.
222 0604
223 0605 RELATIVE_BLOCK = .CURRENT_UCB[UCBSL_RECORD] - .CURRENT_VCB[VCBSL_ST_RECORD];
224 0606
225 0607 IF (.RELATIVE_BLOCK EQL 0 OR .CURRENT_UCB[UCBSL_RECORD] EQLU 1)
226 0608     AND
227 0609     .CURRENT_VCB[VCBSB_TM] EQLU 0
228 0610 THEN
229 0611     RETURN READ_HDR();
230 0612
231 0613 RETURN 1;
232 0614
233 0615 END; ! end of routine

```

```

.TITLE  HEADER
.IDENT  \V04-000\

.EXTRN  CURRENT_UCB, IO_PACKET
.EXTRN  HDR1, HDR2, HDR3
.EXTRN  HDR4, MOUNT_VOL

```

```
.EXTRN LOCAL_FIB
.PSECT $CODE$,NOWRT,2
```

PC	Op	OpC	OpD	OpE	OpF	OpG	OpH	OpI	OpJ	OpK	OpL	OpM	OpN	OpO	OpP	OpQ	OpR	OpS	OpT	OpU	OpV	OpW	OpX	OpY	OpZ	OpAA	OpAB	OpAC	OpAD	OpAE	OpAF	OpAG	OpAH	OpAI	OpAJ	OpAK	OpAL	OpAM	OpAN	OpAO	OpAP	OpAQ	OpAR	OpAS	OpAT	OpAU	OpAV	OpAW	OpAX	OpAY	OpAZ	OpBA	OpBB	OpBC	OpBD	OpBE	OpBF	OpBG	OpBH	OpBI	OpBJ	OpBK	OpBL	OpBM	OpBN	OpBO	OpBP	OpBQ	OpBR	OpBS	OpBT	OpBU	OpBV	OpBW	OpBX	OpBY	OpBZ	OpCA	OpCB	OpCC	OpCD	OpCE	OpCF	OpCG	OpCH	OpCI	OpCJ	OpCK	OpCL	OpCM	OpCN	OpCO	OpCP	OpCQ	OpCR	OpCS	OpCT	OpCU	OpCV	OpCW	OpCX	OpCY	OpCZ	OpDA	OpDB	OpDC	OpDD	OpDE	OpDF	OpDG	OpDH	OpDI	OpDJ	OpDK	OpDL	OpDM	OpDN	OpDO	OpDP	OpDQ	OpDR	OpDS	OpDT	OpDU	OpDV	OpDW	OpDX	OpDY	OpDZ	OpEA	OpEB	OpEC	OpED	OpEE	OpEF	OpEG	OpEH	OpEI	OpEJ	OpEK	OpEL	OpEM	OpEN	OpEO	OpEP	OpEQ	OpER	OpES	OpET	OpEU	OpEV	OpEW	OpEX	OpEY	OpEZ	OpFA	OpFB	OpFC	OpFD	OpFE	OpFF	OpFG	OpFH	OpFI	OpFJ	OpFK	OpFL	OpFM	OpFN	OpFO	OpFP	OpFQ	OpFR	OpFS	OpFT	OpFU	OpFV	OpFW	OpFX	OpFY	OpFZ	OpGA	OpGB	OpGC	OpGD	OpGE	OpGF	OpGG	OpGH	OpGI	OpGJ	OpGK	OpGL	OpGM	OpGN	OpGO	OpGP	OpGQ	OpGR	OpGS	OpGT	OpGU	OpGV	OpGW	OpGX	OpGY	OpGZ	OpHA	OpHB	OpHC	OpHD	OpHE	OpHF	OpHG	OpHH	OpHI	OpHJ	OpHK	OpHL	OpHM	OpHN	OpHO	OpHP	OpHQ	OpHR	OpHS	OpHT	OpHU	OpHV	OpHW	OpHX	OpHY	OpHZ	OpIA	OpIB	OpIC	OpID	OpIE	OpIF	OpIG	OpIH	OpII	OpIJ	OpIK	OpIL	OpIM	OpIN	OpIO	OpIP	OpIQ	OpIR	OpIS	OpIT	OpIU	OpIV	OpIW	OpIX	OpIY	OpIZ	OpJA	OpJB	OpJC	OpJD	OpJE	OpJF	OpJG	OpJH	OpJI	OpJJ	OpJK	OpJL	OpJM	OpJN	OpJO	OpJP	OpJQ	OpJR	OpJS	OpJT	OpJU	OpJV	OpJW	OpJX	OpJY	OpJZ	OpKA	OpKB	OpKC	OpKD	OpKE	OpKF	OpKG	OpKH	OpKI	OpKJ	OpKK	OpKL	OpKM	OpKN	OpKO	OpKP	OpKQ	OpKR	OpKS	OpKT	OpKU	OpKV	OpKW	OpKX	OpKY	OpKZ	OpLA	OpLB	OpLC	OpLD	OpLE	OpLF	OpLG	OpLH	OpLI	OpLJ	OpLK	OpLL	OpLM	OpLN	OpLO	OpLP	OpLQ	OpLR	OpLS	OpLT	OpLU	OpLV	OpLW	OpLX	OpLY	OpLZ	OpMA	OpMB	OpMC	OpMD	OpME	OpMF	OpMG	OpMH	OpMI	OpMJ	OpMK	OpML	OpMM	OpMN	OpMO	OpMP	OpMQ	OpMR	OpMS	OpMT	OpMU	OpMV	OpMW	OpMX	OpMY	OpMZ	OpNA	OpNB	OpNC	OpND	OpNE	OpNF	OpNG	OpNH	OpNI	OpNJ	OpNK	OpNL	OpNM	OpNN	OpNO	OpNP	OpNQ	OpNR	OpNS	OpNT	OpNU	OpNV	OpNW	OpNX	OpNY	OpNZ	OpOA	OpOB	OpOC	OpOD	OpOE	OpOF	OpOG	OpOH	OpOI	OpOJ	OpOK	OpOL	OpOM	OpON	OpOO	OpOP	OpOQ	OpOR	OpOS	OpOT	OpOU	OpOV	OpOW	OpOX	OpOY	OpOZ	OpPA	OpPB	OpPC	OpPD	OpPE	OpPF	OpPG	OpPH	OpPI	OpPJ	OpPK	OpPL	OpPM	OpPN	OpPO	OpPP	OpPQ	OpPR	OpPS	OpPT	OpPU	OpPV	OpPW	OpPX	OpPY	OpPZ	OpQA	OpQB	OpQC	OpQD	OpQE	OpQF	OpQG	OpQH	OpQI	OpQJ	OpQK	OpQL	OpQM	OpQN	OpQO	OpQP	OpQQ	OpQR	OpQS	OpQT	OpQU	OpQV	OpQW	OpQX	OpQY	OpQZ	OpRA	OpRB	OpRC	OpRD	OpRE	OpRF	OpRG	OpRH	OpRI	OpRJ	OpRK	OpRL	OpRM	OpRN	OpRO	OpRP	OpRQ	OpRR	OpRS	OpRT	OpRU	OpRV	OpRW	OpRX	OpRY	OpRZ	OpSA	OpSB	OpSC	OpSD	OpSE	OpSF	OpSG	OpSH	OpSI	OpSJ
----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

; Routine Size: 142 bytes, Routine Base: \$CODES\$ + 0000

: 234 0616 1


```

236 0617 1 GLOBAL ROUTINE READ_HDR : COMMON_CALL =
237 0618 1
238 0619 1 ++
239 0620 1
240 0621 1 FUNCTIONAL DESCRIPTION:
241 0622 1     Read HDR1, and HDR2 if it exists - otherwise, it is defaulted.
242 0623 1     HDR3 is read only if HDR2 is found, and if starlet file. HDR4
243 0624 1     is read if the HDR3 is read.
244 0625 1
245 0626 1 CALLING SEQUENCE:
246 0627 1     READ_HDR()
247 0628 1
248 0629 1 INPUT PARAMETERS:
249 0630 1     none
250 0631 1
251 0632 1 IMPLICIT INPUTS:
252 0633 1     CURRENT_VCB - address of VCB
253 0634 1
254 0635 1 OUTPUT PARAMETERS:
255 0636 1     none
256 0637 1
257 0638 1 IMPLICIT OUTPUTS:
258 0639 1     HDR1, HDR2, HDR3 , and HDR4 read in
259 0640 1     If starlet file, VCB notes this fact
260 0641 1     Also the number of labels that the mtaacp found is set in the VCB
261 0642 1     If logical end of tape (ie: tm encountered on read of HDR1) then this fact is noted in VCB
262 0643 1
263 0644 1 ROUTINE VALUE:
264 0645 1     0 - tm encountered when reading HDR1, logical end of volume set
265 0646 1     1 - successful
266 0647 1
267 0648 1 SIDE EFFECTS:
268 0649 1     First user label may be located in scratch label area
269 0650 1
270 0651 1 USER ERRORS:
271 0652 1     $$$_TAPEPOSLOST - HDR1 not encountered on read
272 0653 1
273 0654 1 --
274 0655 1
275 0656 2 BEGIN
276 0657 2
277 0658 2 LOCAL
278 0659 2     MVL      : REF BBLOCK,
279 0660 2     NUMBER_OF_LABELS,
280 0661 2     SCRATCH : REF BBLOCK,
281 0662 2     DESCR   : VECTOR [2,LONG];
282 0663 2
283 0664 2 EXTERNAL REGISTER
284 0665 2     COMMON_REG;
285 0666 2
286 0667 2 EXTERNAL ROUTINE
287 0668 2     CHCK_IO CLR_EXCP : COMMON_CALL NOVALUE,
288 0669 2     ISSUE_IO       : L$ISSUE_IO,      ! Issue an IO to tape drive
289 0670 2     READ_BLOCK     : COMMON_CALL;    ! read one magtape block
290 0671 2
291 0672 2 BIND
292 0673 2     CVT5 = DESCRIPTOR('!5ZW'),

```

```

293      0674      DEFAULT = UPLIT ('00512');
294      0675
295      0676      ! Initialize the number of labels read. This number will eventually
296      0677      ! be stored in the VCB and will be used on volume switch and file close
297      0678      ! to determine the number of labels to write to the tape
298      0679
299      0680      NUMBER_OF_LABELS = 0;
300      0681      IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
301      0682      THEN
302      0683          BEGIN
303      0684              KERNEL_CALL(UPDVCB_LEOV, 1);
304      0685              RETURN 0;
305      0686
306      0687          END;
307      0688
308      0689      WHILE 1
309      0690      DO
310      0691          BEGIN
311      0692              IF .HDR1[HD1$SL_HD1LID] EQLU 'HDR1'
312      0693              THEN
313      0694                  EXITLOOP;
314      0695
315      0696              IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
316      0697              THEN
317      0698                  ERR_EXIT(SS$_TAPEPOSLOST);
318      0699
319      0700              END;
320      0701
321      0702      NUMBER_OF_LABELS = 1;
322      0703      KERNEL_CALL(MAKE_CUR_FILE, .HDR1);
323      0704
324      0705      ! Default HDR2, HDR3, and HDR4 values
325      0706
326      0707      CH$FILL(' ', ANSI_LBLSZ, .HDR2);
327      0708      CH$FILL(0, ANSI_LBLSZ, .HDR3);
328      0709      CH$FILL(' ', ANSI_LBLSZ, .HDR4);
329      0710
330      0711      ! Default the HDR4 fields according to the version type.
331      0712
332      0713      MVL = .CURRENT_VCB[VCB$MVL];
333      0714      IF .MVL[MVL$B_STDVER] GTR 3
334      0715      THEN
335      0716          HDR4[HD4$B_FILEID_EXT_SIZE] = 0
336      0717          ELSE
337      0718              CH$FILL('0', HD4$_FILEID_EXT_V3, HDR4[HD4$T_FILEID_EXT_V3]);
338      0719
339      0720      HDR2[HD2$B_RECFORMAT] = 'F';
340      0721      DESCR[0] = HD2$_BLOCKLEN;
341      0722      DESCR[1] = HDR2[HD2$T_BLOCKLEN];
342      0723
343      0724      IF NOT $FAO(CVT5, 0, DESCR, .CURRENT_UCB[UCB$_DEVBUFSIZ])
344      0725      THEN
345      0726          CH$MOVE(HD2$_BLOCKLEN, DEFAULT, HDR2[HD2$T_BLOCKLEN]);
346      0727
347      0728      CH$MOVE(HD2$_RECLEN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLEN]);
348      0729
349      0730

```

```

IF .CURRENT_VCB[VCBSW_RECORDSZ] NEQ 0
THEN
  BEGIN
    DESCR[0] = HD2$S_RECLN;
    DESCR[1] = HDR2[HD2$T_RECLN];

    IF NOT $FAO(CVT5, 0, DESCR, .CURRENT_VCB[VCBSW_RECORDSZ])
    THEN
      CH$MOVE(HD2$S_RECLN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLN]);

    END;

    ! Set up the default buffer offset length field. In case there
    ! is no HDR2 label
    HDR2[HD2$T_BUFOFF] = '00';

    ! Set up the Scratch area to read the labels into to determine if
    ! this is a good label, before copying it into the real label field.
    SCRATCH = .HDR1 + SCRATCH_OFFSET;

    ! Now try to read HDR2
    !
    IF READ_BLOCK(.SCRATCH, ANSI_LBLSZ)      ! read into scratch area
    THEN
      IF (.SCRATCH) EQLU 'HDR2'
      THEN
        BEGIN
          CH$MOVE(ANSI_LBLSZ, .SCRATCH, .HDR2);      ! HDR2 found
          NUMBER_OF_LABELS = 2;

          IF .CURRENT_VCB[VCBSV_STARFILE]      ! if starlet file
          THEN
            BEGIN
              IF READ_BLOCK(.SCRATCH, ANSI_LBLSZ)      ! try to read HDR3
              THEN
                BEGIN
                  IF (.SCRATCH) EQLU 'HDR3'
                  THEN
                    BEGIN
                      CH$MOVE(ANSI_LBLSZ, .SCRATCH, .HDR3);      ! HDR3 found
                      NUMBER_OF_LABELS = 3;
                    END;
                  IF READ_BLOCK(.SCRATCH, ANSI_LBLSZ)      ! try to read HDR4
                  THEN
                    IF (.SCRATCH) EQLU 'HDR4'
                    THEN
                      BEGIN
                        CH$MOVE(ANSI_LBLSZ, .SCRATCH, .HDR4);      ! HDR4 found
                        NUMBER_OF_LABELS = 4;
                      END;
                    END;
                  END;
                END;
              END;
            END;
          END;
        END;
      END;
    END;
  END;
END;

```

350 0731
351 0732
352 0733
353 0734
354 0735
355 0736
356 0737
357 0738
358 0739
359 0740
360 0741
361 0742
362 0743
363 0744
364 0745
365 0746
366 0747
367 0748
368 0749
369 0750
370 0751
371 0752
372 0753
373 0754
374 0755
375 0756
376 0757
377 0758
378 0759
379 0760
380 0761
381 0762
382 0763
383 0764
384 0765
385 0766
386 0767
387 0768
388 0769
389 0770
390 0771
391 0772
392 0773
393 0774
394 0775
395 0776
396 0777
397 0778
398 0779
399 0780
400 0781
401 0782
402 0783
403 0784
404 0785
405 0786
406 0787


```

: 407      0788      2
: 408      0789
: 409      0790      ! Call to clear TMSCP drives of the serious exception (reading the tape
: 410      0791      ! mark) before returning to the user
: 411      0792
: 412      0793      CHCK_IO_CLR_EXCP();
: 413      0794      KERNEL_CALLTSET_NUMBER_OF_LABELS, NUMBER_OF_LABELS);
: 414      0795      RETURN-1;                                ! return success
: 415      0796      END;                                    ! end of routine

```

```

57 5A 35 21 0008E P.AAB: .ASCII \!5ZW\
00092 .BLKB 2
00000004 00094 P.AAA: .LONG 4
00000000 00098 .ADDRESS P.AAB
00 00 00 32 31 35 30 30 0009C P.AAC: .ASCII \00512\<0><0><0>

```

```

CVT5=
DEFAULT=
P.AAA
P.AAC
.EXTRN CHCK_IO_CLR_EXCP
.EXTRN ISSUE_IO, READ_BLOCK
.EXTRN SYSSCMKRNL, SYSSFAO

```

			07FC 00000	.ENTRY	READ_HDR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10	0617
5A	0000G	CF	9E 00002	MOVAB	HDR2, R10	
59	00000000G	9F	9E 00007	MOVAB	@SYSSCMKRNL, R9	
5E		08	C2 0000E	SUBL2	#8, SP	
		58	D4 00011	CLRL	NUMBER_OF_LABELS	0680
7E	50	8F	9A 00013	MOVZBL	#80, -(SP)	0681
	0000G	CF	DD 00017	PUSHL	HDR1	
0000G	CF	02	FB 0001B	CALLS	#2, READ_BLOCK	
10		50	E8 00020	BLBS	R0, 1\$	
		01	DD 00023	PUSHL	#1	0684
		01	DD 00025	PUSHL	#1	
	0000V	5E	DD 00027	PUSHL	SP	
69		CF	9F 00029	PUSHAB	UPDVCB LEOV	
		04	FB 0002D	CALLS	#4, SYSSCMKRNL	
31524448	8F	0165	31 00030	BRW	9\$	0685
	0000G	DF	D1 00033	CMPL	@HDR1, #827475016	0693
		16	13 0003C	BEQL	2\$	
7E	50	8F	9A 0003E	MOVZBL	#80, -(SP)	0697
	0000G	CF	DD 00042	PUSHL	HDR1	
0000G	CF	02	FB 00046	CALLS	#2, READ_BLOCK	
E5		50	E8 0004B	BLBS	R0, 1\$	
	0224	8F	BF 0004E	CHMU	#548	0699
		DF	11 00052	BRB	1\$	0689
58		01	DD 00054	MOVL	#1, NUMBER_OF_LABELS	0703
	0000G	CF	DD 00057	PUSHL	HDR1	0704
		01	DD 0005B	PUSHL	#1	
		5E	DD 0005D	PUSHL	SP	
	0000V	CF	9F 0005F	PUSHAB	MAKE_CUR_FILE	
69		04	FB 00063	CALLS	#4, SYSSCMKRNL	
0050	8F	20	6A DD 00066	MOVL	HDR2, R7	0708
		6E	00 2C 00069	MOVCS	#0, (SP), #32, #80, (R7)	
		67	00 2C 00070			
0050	8F	00	6E 00 2C 00071	MOVCS	#0, (SP), #0, #80, @HDR3	0709

	0000G	CF		57	DD	00164	PUSHL	SCRATCH	
		14		02	FB	00166	CALLS	#2, READ_BLOCK	
	34524448	8F		50	E9	00168	BLBC	R0, 8\$	
				67	D1	0016E	CMPL	(SCRATCH), #877806664	0779
0000G	DF			0B	12	00175	BNEQ	8\$	
		67	0050	8F	28	00177	MOVCS	#80, (SCRATCH), @HDR4	0782
	0000G	CF		04	D0	0017F	MOVL	#4, NUMBER OF LABELS	0783
				00	FB	00182	CALLS	#0, CHCK_ID CLR_EXCP	0792
				58	DD	00187	PUSHL	NUMBER_OF_LABELS	0793
				01	DD	0C189	PUSHL	#1	
				SE	DD	0018B	PUSHL	SP	
			0000V	CF	9F	0018D	PUSHAB	SET_NUMBER OF LABELS	
		69		04	FB	00191	CALLS	#4, -SYS\$CMRNC	
		50		01	D0	00194	MOVL	#1, R0	0794
					04	00197	RET		
				50	D4	00198	CLRL	R0	0796
				04	0019A	RET			

; Routine Size: 411 bytes, Routine Base: \$CODE\$ + 00A4

; 416 0797 1


```

418 0798 1 GLOBAL ROUTINE WRAP_AROUND : L$WRAP_AROUND =
419 0799 1
420 0800 1 ++
421 0801 1
422 0802 1 FUNCTIONAL DESCRIPTION:
423 0803 1     If this is not the first time through and the search started
424 0804 1     at the beginning of the volume set then return error else rewind volume set
425 0805 1
426 0806 1 CALLING SEQUENCE:
427 0807 1     WRAP_AROUND()
428 0808 1
429 0809 1 INPUT PARAMETERS:
430 0810 1     none
431 0811 1
432 0812 1 IMPLICIT INPUTS:
433 0813 1     LOCAL_FIB - copy of user's fib
434 0814 1     CURRENT_VCB - address of current volume VCB
435 0815 1
436 0816 1 OUTPUT PARAMETERS:
437 0817 1     none
438 0818 1
439 0819 1 IMPLICIT OUTPUTS:
440 0820 1     none
441 0821 1
442 0822 1 ROUTINE VALUE:
443 0823 1     0 back to beginning of search
444 0824 1     1 at beginning of volume set
445 0825 1
446 0826 1 SIDE EFFECTS:
447 0827 1     none
448 0828 1
449 0829 1 --
450 0830 1
451 0831 2 BEGIN
452 0832 2
453 0833 2 EXTERNAL REGISTER
454 0834 2     COMMON_REG;
455 0835 2
456 0836 2 EXTERNAL ROUTINE
457 0837 2     MOUNT_VOL      : COMMON_CALL,      ! mount volume
458 0838 2     REWIND_VOL_SET : COMMON_CALL;      ! rewind volume set
459 0839 2
460 0840 2 EXTERNAL
461 0841 2     LOCAL_FIB      : BBLOC      ! copy of user's fib
462 0842 2
463 0843 2 IF .CURRENT_VCB[VCB$START_FID] EQL 'X'00010001'
464 0844 2 THEN
465 0845 2     RETURN 0
466 0846 2 ELSE
467 0847 2     BEGIN
468 0848 2         REWIND_VOL_SET();
469 0849 2
470 0850 2         ! get first volume mounted
471 0851 2
472 0852 2         MOUNT_VOL(1, $FIELDMASK(MOUSV_REWIND) + $FIELDMASK(MOUSV_LBLCHECK));
473 0853 2
474 0854 2     IF NOT READ_HDR()

```

```

: 475      0855
: 476      0856
: 477      0857
: 478      0858
: 479      0859
: 480      0860
: 481      0861
: 482      0862

```

```

      THEN
      ERR_EXIT(SS$_TAPEPOSLOST);

      END;

      RETURN 1;

      END;

```

! end of routine

.EXTRN REWIND_VOL_SET

```

00010001  8F      28  AB  D1 00000 WRAP_AROUND::
                                CMPL  40(CURRENT_VCB), #65537
                                BEQL  2$
                                0000G  CF      00  FB 0000A  CALLS  #0, REWIND_VOL_SET
                                03  DD 0000F  PUSHL  #3
                                01  DD 00011  PUSHL  #1
                                0000G  CF      02  FB 00013  CALLS  #2, MOUNT_VOL
                                FE48  CF      00  FB 00018  CALLS  #0, READ_RDR
                                04      50  E8 0001D  BLBS  R0, 1$
                                0224  8F  BF 00020  CHMU  #548
                                50      01  D0 00024 1$:  MOVL  #1, R0
                                05 00027  RSB
                                50  D4 00028 2$:  CLRL  R0
                                05 0002A  RSB

```

; Routine Size: 43 bytes, Routine Base: \$CODE\$ + 023F

; 483 0863 1

```

485 0864 1 GLOBAL ROUTINE SPACE_EOF : COMMON_CALL NOVALUE =
486 0865 1
487 0866 1 ++
488 0867 1
489 0868 1 FUNCTIONAL DESCRIPTION:
490 0869 1     This routine spaces to the end of the current file, right
491 0870 1     before the next file.
492 0871 1
493 0872 1 CALLING SEQUENCE:
494 0873 1     SPACE_EOF()
495 0874 1
496 0875 1 INPUT PARAMETERS:
497 0876 1     none
498 0877 1
499 0878 1 IMPLICIT INPUTS:
500 0879 1     CURRENT_VCB _ address of current VCB
501 0880 1
502 0881 1 OUTPUT PARAMETERS:
503 0882 1     none
504 0883 1
505 0884 1 IMPLICIT OUTPUTS:
506 0885 1     none
507 0886 1
508 0887 1 ROUTINE VALUE:
509 0888 1     none
510 0889 1
511 0890 1 SIDE EFFECTS:
512 0891 1     The tape is left positioned in front of HDR1 of the next file
513 0892 1
514 0893 1 --
515 0894 1
516 0895 2 BEGIN
517 0896 2
518 0897 2 SWITCHES NOOPTIMIZE;
519 0898 2
520 0899 2 EXTERNAL REGISTER
521 0900 2     COMMON_REG;
522 0901 2
523 0902 2 EXTERNAL ROUTINE
524 0903 2     GTNEXT_VOL_READ : JSB,           ! get next volume on read
525 0904 2     READ_BLOCK      : COMMON_CALL,   ! read mag tape block
526 0905 2     SPACE_TM        : COMMON_CALL;    ! space tm's
527 0906 2
528 0907 2 EXTERNAL
529 0908 2     CURRENT_UCB      : REF BBLOCK;      ! address of current ucb
530 0909 2
531 0910 2 LOCAL
532 0911 2     TM;
533 0912 2
534 0913 2 ! If tape is positioned in header set, space 2 tape marks
535 0914 2 !
536 0915 2
537 0916 2 IF .CURRENT_VCB[VCBSB_TM] EQL 0 AND .HDR1[HD1$SL_HD1LID] EQL 'HDR1'
538 0917 2 THEN
539 0918 2     SPACE_TM(2);
540 0919 2
541 0920 2 ! if in data area, space 1 tape mark

```



```

542 0921 !
543 0922
544 0923 IF .CURRENT_VCB[VCB$B_TM] EQLU 1
545 0924 THEN
546 0925     SPACE_TM(1);
547 0926
548 0927 ! Now if trailer label has not been read, read it
549 0928 !
550 0929
551 0930 IF .CURRENT_VCB[VCB$B_TM] EQLU 2
552 0931     AND
553 0932     (.CURRENT_UCB[UCB$L_RECORD] - .CURRENT_VCB[VCB$L_ST_RECORD]) EQL 0
554 0933 THEN
555 0934
556 0935     IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
557 0936     THEN
558 0937         ERR_EXIT(SS$_TAPEPOSLOST);
559 0938
560 0939 WHILE 1
561 0940 DO
562 0941     BEGIN
563 0942
564 0943     IF .HDR1[HD1$L_HD1LID] EQL 'EOF1'
565 0944     THEN
566 0945         EXITLOOP;
567 0946
568 0947     IF .HDR1[HD1$L_HD1LID] NEQ 'EOV1'
569 0948     THEN
570 0949         ERR_EXIT(SS$_TAPEPOSLOST);
571 0950
572 0951     GTNEXT_VOL_READ();
573 0952
574 0953     IF .CURRENT_VCB[VCB$B_TM] EQLU 0
575 0954     THEN
576 0955         SPACE_TM(2)
577 0956     ELSE
578 0957         SPACE_TM(1);
579 0958
580 0959     IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
581 0960     THEN
582 0961         ERR_EXIT(SS$_TAPEPOSLOST);
583 0962
584 0963     END;
585 0964
586 0965 IF .CURRENT_VCB[VCB$B_TM] EQLU 2
587 0966 THEN
588 0967     SPACE_TM(1);
589 0968
590 0969 END;

```

! end of routine

.EXTRN GTNEXT_VOL_READ
.EXTRN SPACE_TM

52 0000G CF 07FC 0000
9E 00002

.ENTRY SPACE_EOF, Save R2,R3,R4,R5,R6,R7,R8,R9,R10 : 0864
MOVAB SPACE_TM, R2

		2E	AB	95	00007	TSTB	46(CURRENT_VCB)	0916
			10	12	0007A	BNEQ	1\$	
31524448	8F	0000G	0F	D1	0000C	CMPL	@HDR1, #827475016	
			05	12	00015	BNEQ	1\$	
	62		02	DD	00017	PUSHL	#2	0918
	01		01	FB	00019	CALLS	#1, SPACE_TM	
		2E	AB	91	0001C	1\$: CMPB	46(CURRENT_VCB), #1	0923
			05	12	00020	BNEQ	2\$	
			01	DD	00022	PUSHL	#1	0925
	62		01	FB	00024	CALLS	#1, SPACE_TM	
	02		2E	AB	91	00027	2\$: CMPB	46(CURRENT_VCB), #2
			21	12	0002B	BNEQ	4\$	0930
	50	0000G	CF	D0	0002D	MOVL	CURRENT_UCB, R0	0932
30	AB	00B0	C0	D1	00032	CMPL	176(R0), 48(CURRENT_VCB)	
			14	12	00038	BNEQ	4\$	
	7E		50	8F	9A	0003A	3\$: MOVZBL	#80, -(SP)
		0000G	CF	DD	0003E	PUSHL	HDR1	0935
	0000G	CF	02	FB	00042	CALLS	#2, READ_BLOCK	
			50	EB	00047	BLBS	R0, 4\$	
		0224	8F	BF	0004A	CHMU	#548	0937
31464F45	8F	0000G	DF	D1	0004E	4\$: CMPL	@HDR1, #826691397	0943
			22	13	00057	BEQL	8\$	
31564F45	8F	0000G	DF	D1	00059	CMPL	@HDR1, #827739973	0947
			04	13	00062	BEQL	5\$	
		0224	8F	BF	00064	CHMU	#548	0949
		0000G	30	00068	5\$: BSBW	GTNEXT VOL_READ		0951
		2E	AB	95	0006B	TSTB	46(CURRENT_VCB)	0953
			04	12	0006E	BNEQ	6\$	
			02	DD	00070	PUSHL	#2	0955
			02	11	00072	BRB	7\$	
	62		01	DD	00074	6\$: PUSHL	#1	0957
			01	FB	00076	7\$: CALLS	#1, SPACE_TM	
			BF	11	00079	BRB	3\$	0959
	02		2E	AB	91	0007B	8\$: CMPB	46(CURRENT_VCB), #2
			05	12	0007F	BNEQ	9\$	0965
			01	DD	00081	PUSHL	#1	0967
	62		01	FB	00083	CALLS	#1, SPACE_TM	
			04	00086	9\$: RET			0969

; Routine Size: 135 bytes, Routine Base: \$CODE\$ + 026A

; 591 0970 1

```

593 0971 1 ROUTINE MAKE_CUR_FILE (LABELS) : COMMON_CALL NOVALUE =
594 0972 1
595 0973 1 ++
596 0974 1
597 0975 1 FUNCTIONAL DESCRIPTION:
598 0976 1 This routine updates the current file number and the Starlet
599 0977 1 file indicator.
600 0978 1
601 0979 1 CALLING SEQUENCE:
602 0980 1 MAKE_CUR_FILE(ARG1), call in kernel mode
603 0981 1
604 0982 1 INPUT PARAMETERS:
605 0983 1 ARG1 - address of labels
606 0984 1
607 0985 1 IMPLICIT INPUTS:
608 0986 1 none
609 0987 1
610 0988 1 OUTPUT PARAMETERS:
611 0989 1 none
612 0990 1
613 0991 1 IMPLICIT OUTPUTS:
614 0992 1 If file is Starlet file, then VCB$V_STARFILE = 1
615 0993 1 CUR_NUM is updated
616 0994 1
617 0995 1 ROUTINE VALUE:
618 0996 1 none
619 0997 1
620 0998 1 SIDE EFFECTS:
621 0999 1 none
622 1000 1
623 1001 1 --
624 1002 1
625 1003 2 BEGIN
626 1004 2
627 1005 2 EXTERNAL REGISTER
628 1006 2 COMMON_REG;
629 1007 2
630 1008 2 MAP
631 1009 2 LABELS : REF BBLOCK; ! HDR1, HDR2, and HDR3 address
632 1010 2
633 1011 2 BIND
634 1012 2
635 1013 2 ! Any file with 11 code will be supported, instead of only 11A
636 1014 2
637 1015 2 STARID = UPLIT ('DECFILE11');
638 1016 2
639 1017 2 EXTERNAL ROUTINE
640 1018 2 FORMAT_FID : COMMON_CALL; ! format file id
641 1019 2
642 1020 2 CURRENT_VCB[VCB$V_STARFILE] = CH$EQL(9, STARID, 9, LABELS[HD1$T_SYSCODE],0);
643 1021 2 FORMAT_FID(CURRENT_VCB[VCB$W_CUR_NUM]);
644 1022 1 END; ! end of routine

```

```

00 00 00 31 31 45 4C 49 46 43 45 44 002F1 .BLKB 3
002F4 P.AAD: .ASCII \DECFILE11\<0><0><0>

```


STARID= .EXTRN P.AAD
FORMAT_FID

001C 00000 MAKE_CUR_FILE:

			50	04	AC	D0	00002		WORD	Save R2,R3,R4		0971
					54	D4	00006		MOVL	LABELS, R0		1020
	3C	A0	E8	AF	09	29	00008		CLRL	R4		
					02	12	0000E		CMPC3	#9, STARID, 60(R0)		
					54	D6	00010		BNEQ	1\$		
2D	AB		01	00	54	F0	00012	1\$:	INCL	R4		
					54	F0	00012		INSV	R4, #0, #1, 45(CURRENT_VCB)		
					AB	9F	00018		PUSHAB	36(CURRENT_VCB)		1021
			0000G	CF	01	FB	0001B		CALLS	#1, FORMAT_FID		
					04	00020			RET			1022

; Routine Size: 33 bytes, Routine Base: \$CODE\$ + 0300

; 645 1023 1

```

647 1024 1 GLOBAL ROUTINE UPDVCB_LEOV (BIT_VALUE) : COMMON_CALL NOVALUE =
648 1025 1
649 1026 1 ++
650 1027 1
651 1028 1 FUNCTIONAL DESCRIPTION:
652 1029 1     This routine sets or clears the logical end of file bit in the VCB
653 1030 1
654 1031 1 CALLING SEQUENCE:
655 1032 1     UPDVCB_LEOV(ARG1), called in kernel mode
656 1033 1
657 1034 1 INPUT PARAMETERS:
658 1035 1     value to set logical end of volume to:
659 1036 1         0 - clear bit
660 1037 1         1 - set bit
661 1038 1
662 1039 1 IMPLICIT INPUTS:
663 1040 1     CURRENT_VCB - address of volume control block
664 1041 1
665 1042 1 OUTPUT PARAMETERS:
666 1043 1     none
667 1044 1
668 1045 1 IMPLICIT OUTPUTS:
669 1046 1     CURRENT_VCB[VCB$V_LOGICEOVS] is set or cleared
670 1047 1
671 1048 1 ROUTINE VALUE:
672 1049 1     none
673 1050 1
674 1051 1 SIDE EFFECTS:
675 1052 1     none
676 1053 1
677 1054 1 --
678 1055 1
679 1056 2 BEGIN
680 1057 2
681 1058 2 EXTERNAL REGISTER
682 1059 2     COMMON_REG;
683 1060 2
684 1061 2 CURRENT_VCB[VCB$V_LOGICEOVS] = .BIT_VALUE;
685 1062 1 END; ! end of routine

```

OB	AB	01	01	04	AC	0000 0000	.ENTRY UPDVCB_LEOV, Save nothing	: 1024
						F0 00002	INSV BIT_VALUE, #1, #1, 11(CURRENT_VCB)	: 1061
						04 00009	RET	: 1062

; Routine Size: 10 bytes, Routine Base: \$CODE\$ + 0321

; 686 1063 1

```

: 688 1064 1 ROUTINE SET_NUMBER_OF_LABELS (NUMBER_OF_LABELS) : COMMON_CALL NOVALUE =
: 689 1065 1
: 690 1066 1 ++
: 691 1067 1
: 692 1068 1 FUNCTIONAL DESCRIPTION:
: 693 1069 1 This routine sets then number of labels read by the MTAACP in the VCB.
: 694 1070 1 This value will be used to determine how many labels are written out
: 695 1071 1 won volume switch or at end of file processing. The reason this is
: 696 1072 1 necessary is so that if a file is open with fewer labels then we support
: 697 1073 1 we do not write the greater number of LDR labels out to the tape. This
: 698 1074 1 would be a noncompliance with the ANSI standard for tape label
: 699 1075 1 processing.
: 700 1076 1
: 701 1077 1 CALLING SEQUENCE:
: 702 1078 1 SET_NUMBER_OF_LABELS(ARG1), called in kernel mode
: 703 1079 1
: 704 1080 1 INPUT PARAMETERS:
: 705 1081 1 Number of labels read.
: 706 1082 1
: 707 1083 1 IMPLICIT INPUTS:
: 708 1084 1 CURRENT_VCB - address of volume control block
: 709 1085 1
: 710 1086 1 OUTPUT PARAMETERS:
: 711 1087 1 none
: 712 1088 1
: 713 1089 1 IMPLICIT OUTPUTS:
: 714 1090 1 CURRENT_VCB[VCB$B_LBLCNT] is set
: 715 1091 1
: 716 1092 1 ROUTINE VALUE:
: 717 1093 1 none
: 718 1094 1
: 719 1095 1 SIDE EFFECTS:
: 720 1096 1 none
: 721 1097 1
: 722 1098 1 --
: 723 1099 1
: 724 1100 2 BEGIN
: 725 1101 2
: 726 1102 2 EXTERNAL REGISTER
: 727 1103 2 COMMON_REG;
: 728 1104 2
: 729 1105 2 CURRENT_VCB[VCB$B_LBLCNT] = .NUMBER_OF_LABELS;
: 730 1106 1 END; ! end of routine

```

0000 00000 SET_NUMBER_OF_LABELS:

48	AB	04	AC	90 00002	WORD	Save nothing	:	1064
				04 00007	MOVB	NUMBER_OF_LABELS, 72(CURRENT_VCB)	:	1105
					RET		:	1106

; Routine Size: 8 bytes, Routine Base: \$CODE\$ + 032B

; 731 1107 1

HEADER
V04-000

J 2
16-Sep-1984 02:22:07
14-Sep-1984 12:46:41

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]HEADER.B32;1

Page 22
(8)

: 732 1108 1 END
: 733 1109 1
: 734 1110 0 ELUDOM

PSECT SUMMARY

: Name Bytes Attributes
: \$CODE\$ 819 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _\$255\$DUA28:[SYSLIB]LIB.L32;1 18619 36 0 1000 00:01.8

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:HEADER/OBJ=OBJ\$:HEADER MSRC\$:HEADER/UPDATE=(ENH\$:HEADER)

: Size: 782 code + 37 data bytes
: Run Time: 00:17.7
: Elapsed Time: 00:40.6
: Lines/CPU Min: 3771
: Lexemes/CPU-Min: 18091
: Memory Used: 163 pages
: Compilation Complete

0254 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0255

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY